

# Studi dan Penerapan *Side Channel Attack: Timing Attack*

Sandy Socrates / 13508044<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia

if18044@students.if.itb.ac.id

**Abstract** — Zaman terus berkembang, termasuk juga kriptografi. Seiring dengan perkembangan kriptografi, para kriptanalis juga mengembangkan serangan-serangan baru untuk mengeksploitasi informasi yang disimpan melalui proses kriptografi. Mulai awal tahun 1990-an muncul pendekatan baru untuk mendapatkan informasi yang terenkripsi, pendekatan ini tidak menggunakan bruteforce atau kelemahan teoritis dari sebuah sistem kriptografi, namun menggunakan efek tidak langsung dari sebuah proses kriptografi, pendekatan ini adalah *side channel attack*. Contoh jenis serangan ini meliputi *timing attack*, *power monitoring attack*, dan *electromagnetic attack*. Pada *timing attack*, dimanfaatkan waktu yang digunakan oleh sistem kriptografi sebagai bahan analisa. Mengingat setiap operasi komputer memerlukan waktu eksekusi maka *timing attack* memiliki rentang penerapan yang luas dan memiliki potensi menjadi serangan yang sangat sulit dihadapi.

**Index Terms**— *electromagnetic attack*, *power monitoring attack*, serangan kriptografi, *side channel attack*, *timing attack*.

## I. PENDAHULUAN

Keamanan adalah topik yang sangat krusial dalam penghantaran informasi. Adanya rasa takut antar pihak yang akan bertukar informasi membuat munculnya sistem-sistem kriptografi. Akan tetapi hal itu juga menimbulkan ketakutan baru bagi pengirim pesan yaitu para kriptanalis yang “penasaran” dengan isi informasi yang akan ditukarkan.

Berbicara mengenai keamanan memang tidak akan pernah habis. Misalkan saja jika kita membicarakan enkripsi-dekripsi, selalu saja ada adu antara kriptografer dan kriptanalis untuk saling mengungguli satu sama lain. Setiap kali dibuat sebuah sistem kriptografi, dibuat juga sebuah cara untuk mendapatkan informasi di dalamnya.

Akan tetapi keamanan dari sebuah sistem memang sulit untuk dinilai. Seringkali tertanam sebuah konsep bahwa dengan enkripsi yang bagus maka

keamanan dari sebuah transaksi informasi sudah terjamin. Padahal bisa saja seorang kriptanalis mendapatkan informasi yang diinginkan bahkan tanpa menyentuh sistem kriptografi yang digunakan.

Bayangkan anda memiliki sebuah rumah dengan barang yang berharga untuk anda di dalamnya. Untuk melindungi barang tersebut anda membeli kunci dan gembok paling mahal di kota anda. Dengan santai maka anda dapat pergi meninggalkan rumah tanpa mengkhawatirkan kondisi rumah dan barang anda. Akan tetapi bisa saja ada pencuri yang tidak perlu melihat kunci pintu apa yang anda gunakan dan dengan mudah dan sederhana melepas sekrup pintu tanpa harus menyentuh kunci. Dengan itu pencuri bisa membuka pintu bahkan tanpa sempat melihat merek kunci apa yang anda gunakan.

Pendekatan tidak langsung seperti di atas adalah ilustrasi dari *side channel attack* pada kriptografi. Sedikit mengenai *side channel attack*, jenis serangan ini memanipulasi informasi sampingan dalam proses komputasi pada sistem kriptografi seperti waktu yang digunakan, daya yang terpakai, dan suara yang dihasilkan. Ide dari pemikiran ini benar-benar jenius, karena kita hanya membutuhkan informasi sampingan yang biasanya pengawasannya tidak diperhatikan.

## II. PRINSIP DASAR

### A. *Side Channel Attack*

Sesuai dengan namanya, “*Side Channel Attack*”, adalah serangan yang berdasarkan pada “informasi sampingan”. Informasi sampingan adalah informasi yang dapat diambil dari mesin enkripsi yang bisa berupa plainteks yang akan dienkripsi atau pun cipherteks dari proses dekripsi.[1]

Jauh ke tahun-tahun di belakang, mesin enkripsi adalah mesin yang memiliki fungsi enkripsi dan dekripsi, sehingga mesin dapat merubah plainteks ke cipherteks dan sebaliknya. Maka dari itu, serangan

hanya bisa dilakukan dengan mengetahui salah satu teks tadi atau keduanya. Sekarang, bukan hanya plainteks dan cipherteks yang dihasilkan oleh mesin enkripsi, tapi juga informasi tambahan yang bisa diukur seperti waktu yang digunakan, daya yang dipakai, dan sebagainya. *Side Channel Attack* memanfaatkan informasi-informasi ini untuk mendapatkan informasi utama, yaitu si plainteks.

Pembagian yang umum dari *side channel attack*:

1. *Timing Attack* – serangan yang berdasarkan pengukuran waktu yang digunakan untuk mengeksekusi komputasi yang berbeda-beda
2. *Power Monitoring Attack* – serangan yang menggunakan jumlah konsumsi daya oleh komputer selama melakukan komputasi.
3. *Electromagnetic Attack* – serangan yang berdasarkan radiasi elektromagnetik yang bocor, yang dalam hal ini bisa langsung memberikan plainteks dan informasi lain.
4. *Acoustic Cryptanalysis* – serangan yang mengeksploitasi suara yang dihasilkan selama komputasi.
5. *Cache Attack* - serangan yang mengeksploitasi hubungan antar proses pada komputer.
6. *Differential fault analysis*- serangan yang dilakukan dengan mengeksploitasi kesalahan selama komputasi.

### B. Timing Attack

*Timing attack* bekerja dengan melakukan pengukuran terhadap waktu yang dibutuhkan sebuah sistem kriptografi untuk melakukan sebuah operasi. Informasi ini dapat menuntun kriptanalisis pada informasi mengenai kunci rahasia. Sebagai contoh, jika kita melakukan pengukuran terhadap waktu yang diperlukan untuk melakukan operasi kunci privat, kriptanalisis dapat menemukan eksponen dari Diffie-Hillman, faktor kunci RSA, dan memecahkan sistem kriptografi lainnya[2]. Jika sebuah sistem bisa diserang, maka serangan tersebut sederhana secara komputasi.

Pada sebuah sistem kriptografi, terdapat perbedaan waktu yang dibutuhkan untuk jenis masukan yang berbeda. Terdapat banyak hal yang dapat mempengaruhi kecepatan atau waktu yang diperlukan untuk mengeksekusi operasi, seperti optimasi, beban prosesor, jenis perintah yang dilakukan, dan kondisi mesin saat eksekusi.

Informasi mengenai waktu dapat membawa kita

ke dalam informasi yang penting. Walaupun secara nalar kita berpikir bahwa informasi waktu tidak akan membawa kita ke mana pun.

#### B.1. Prinsip

*Timing attack* adalah contoh dari serangan yang mengeksploitasi kekuatan implementasi dari sebuah algoritma dari pada algoritmanya itu sendiri. Jika algoritma yang sama diimplementasikan ulang sehingga waktu yang diperlukan untuk mengeksekusi sebuah proses selalu sama dengan cara memberikan sebuah konstanta delay pada program. Pada kasus seperti itu *timing attack* tidak bisa menganalisa waktu yang dihasilkan, walaupun hal itu harus dibayar dengan performa yang berkurang karena delay .

Pada pengukuran waktu, dalam pengolahannya menggunakan statistik untuk dicari hubungannya dengan kunci yang dicari.

Karena digunakan statistik diperlukan sejumlah sampel dan pengujian. Jumlah data yang diperlukan sangat bergantung pada variansi sampel, noise, dan kuat sinyal. Semakin banyak noise semakin banyak data yang dibutuhkan. Pada umumnya, teknik pengecekan kesalahan akan menambah kebutuhan memori dan proses untuk melakukan serangan, namun hal ini dapat mengurangi jumlah sampel yang dibutuhkan.[2]

#### B.2. Contoh penggunaan

*Timing attack* terus berkembang, dan telah berhasil digunakan terhadap implementasi SSL (*Secure Socket Layer*) pada jaringan pada tahun 2003 oleh Boneh dan Brumley. Walaupun jaringan yang diserang kecil, namun dalam waktu beberapa jam mereka berhasil mendapatkan *private key* pada server.

Beberapa versi Unix menggunakan implementasi *hash* yang mengubah password 8-karakter menjadi string dengan 11-karakter. Pada hardware yang lama proses eksekusi *hash* ini membutuhkan waktu yang lama, dan pada versi UNIX yang lama fungsi ini hanya akan dipanggil jika *username* yang dimasukkan benar. Hal ini akan membantu penyerang mencari *username* apa yang terdaftar. Perbandingan waktu fungsi ini linear terhadap jumlah karakter password yang benar dari kiri. Jadi penyerang dapat menduga-duga password apa yang digunakan dengan membandingkan waktu yang tercatat. Versi UNIX setelahnya telah memperbaiki kelemahan ini.

### III. TIMING ATTACK PADA RSA

#### A. Latar Belakang

Kocher [2] adalah orang yang pertama kali mendiskusikan *timing attack*. Pada tahun 1996 Kocher mempresentasikan hasil kerjanya pada konferensi *RSA Data Security and CRYPTO* untuk memperingatkan orang-orang mengenai hasil kerjanya. Hal ini mengejutkan para kriptografer termasuk pencipta RSA. Dari presentasi itu dia memperingatkan bahwa informasi samping yang dihasilkan saat proses enkripsi-dekripsi juga berbahaya jika jatuh ke tangan kriptanalis.

#### B. Mengenai RSA

RSA adalah sistem kriptografi kunci publik yang menggunakan sebuah eksponen  $e$  publik untuk enkripsi dan sebuah eksponen  $d$  privat untuk dekripsi.

RSA menggunakan modulus  $N$  yang merupakan hasil kali dari dua bilangan prima  $p$  dan  $q$ . Eksponen  $e$  dan  $d$  harus dipilih untuk memenuhi kondisi

$$ed = 1 \text{ mode } (p - 1)(q - 1).$$

Lalu pasangan kunci RSA terdiri dari kunci publik  $(N, e)$  and dan sebuah kunci private  $d$ . Sebagai contoh, jika kita memilih dua bilangan prima

$$p = 11 \\ q = 3,$$

maka,

$$N = 11 * 3 = 33.$$

Kemudian hitung

$$(p - 1)(q - 1) = 10 * 2 = 20$$

Dan pilih sebuah nilai  $e$  yang relatif prima terhadap 20, misalkan 3. Kemudian  $d$  harus dipilih sedemikian hingga

$$ed = 1 \text{ mod } 20$$

Salah satu nilai yang mungkin untuk  $d$  adalah 7 karena

$$3 * 7 = 21 = 1 \text{ mod } 20.$$

Jadi kita telah memiliki kunci publik  $(N= 33, e= 3)$  dan sebuah kunci privat  $d = 7$ . Kita hilangkan faktor awal yaitu  $p$  dan  $q$ . Faktorisasi dapat memecahkan RSA karena jika seorang penyerang dapat memfaktorkan  $N$  ke dalam  $p$  dan  $q$ , dia dapat menggunakan kunci publik  $e$  untuk mencari kunci private  $d$ .

Untuk mengenkripsi sebuah pesan  $M$  kita menghitung

$$C = M^e \text{ mod } N,$$

Di mana  $C$  adalah pesan yang dienkripsi. Untuk mendekrip pesan  $C$ , kita menghitung

$$M = C^d \text{ mod } N,$$

Yang akan membawa kita ke pesan semula yaitu  $M$ . Jikk dihubungkan dengan contoh kita yang pertama di mana kunci publik  $(N= 33, e= 3)$  dan kunci private  $d= 7$ . Maka ketika kita akan mengirim pesan  $M= 19$ . Enkripsi akan memberikan cipherteks

$$C = M^e \text{ mod } N = 19^3 \text{ mod } 33 = 28.$$

Pengirim akan mengirimkan pesan  $C = 28$ . Untuk mendekripsi cipherteks  $C$ , penerima menggunakan kunci privat  $d$  dan menghitung

$$C^d \text{ mod } N = 28^7 \text{ mod } 33 = 19,$$

Hasil dari operasi diatas merupakan  $M$ . Dekripsi tadi menggunakan Teorema Euler. Kegunaan lain dari RSA adalah sebagai *signature* untuk memverifikasi pengirim pesan. [3]

#### B. Timing Attack

Operasi pada RSA yang menggunakan kunci privat adalah modulo dari perpangkatan

$$M = C^d \text{ mod } N,$$

Tujuan penyerang adalah mencari  $d$ . Untuk sebuah *timing attack*, penyerang harus menyuruh sistem menghitung  $C^d \text{ mod } N$  untuk beberapa nilai  $C$

yang sudah dipilih. Dengan mengukur secara tepat jumlah waktu yang dibutuhkan dan menganalisa variasi waktu, penyerang dapat menemukan kunci privat  $d$  bit-per-bit sampai seluruh perpangkatan diketahui.

Sebagaimana dijelaskan oleh Kocher [2], serangan ini bergantung pada pendeteksian sinyal. Sinyal adalah variasi waktu yang dikarenakan oleh perpangkatan oleh target, sedangkan *noise* ketidakakuratan pengukuran waktu *oleh*.

```

x = C
for j = 1 to n
  x = mod(x2, N)
  if dj == 1 then
    x = mod(xC, N)
  end if
next j
return x

```

Modulo perpangkatan pada RSA adalah operasi yang memakan waktu sangat besar. Algoritma sederhana dan efisien untuk melakukannya ditunjukkan di atas.

Pada umumnya digunakan algoritma Montgomery untuk melakukan perkalian dan kuadrat. Dan cara lain yang digunakan untuk menghitung modulonya adalah *Chinese Remainder Theorem* untuk melakukan perpangkatan[3].

### B.1 Implementasi

Penggunaan RSA paling populer adalah pada smart card. Dham et al. Menggunakan serangan ini untuk versi awal dari *smart card* CASCADE. Kenyataan bahwa smart card ini bisa dimiliki oleh si penyerang membuatnya relatif mudah untuk diukur waktu jalan operasi kriptografi dan maka dari itu *smart card* bisa diserang dengan *timing attack*.

Implementasi Dhem memperhatikan operasi kuadrat daripada perkalian dengan mengamati tambahan pengurangan. Mereka mengakui telah mendapatkan kunci 512 bit dalam beberapa menit dengan pengukuran waktu 350.000. Pada kunci 128-bit dapat dipecahkan dalam waktu 4-bit per detik dengan 10.000 sampel. Pada saat mereka tidak mendapatkan sampel yang cukup untuk mendapatkan kunci yang lengkap mereka masih dapat mendapatkan sebagian kunci. Dhem dan kawan-kawan bahwa dengan hanya setengah dari jumlah sampel yang dibutuhkan mereka mampu memecahkan 75% kunci.

Sistem yang menggunakan *Chinese Remainder Theorem* tidak dapat ditembus oleh metode Kocher dan Dhem. Akan tetapi Brumley dan Boneh berhasil mengimplementasikan *timing attack* pada OpenSSL. Mereka telah dengan sukses mengambil  $N$  dan  $d$  dari RSA. Sebelum hal ini dipublikasikan tidak ada yang percaya bahwa hal ini bisa dilakukan karena waktu pada sebuah server akan tertutup oleh variasi waktu. Mereka berhasil mendapatkan kunci 1024-bit dalam waktu 2 jam dari OpenSSL).9.7.

Serangan-serangan tadi menggambarkan bahwa sistem kriptografi yang bergantung pada perpangkatan modulo bisa diserang dengan *timing attack*. Mereka juga membuktikan bahwa CRT juga bisa dipecahkan dengan *timing attack*.

### B.1 Menghindari Timing Attack

Ada banyak cara yang bisa digunakan untuk menghindari *timing attack*, berikut adalah beberapa di antaranya:

1. *RSA blinding*, dengan sistem ini diberikan sebuah *randomness* pada operasi di RSA sehingga informasi waktu menjadi tidak berguna. Sebelum melakukan dekripsi kita kalikan dengan sebuah bilangan random  $r^{ed}$  kemudian pada saat dekripsi dikalikan dulu dengan  $r^{-1}$ . Dengan cara ini didapatkan waktu yang berbeda-beda namun hasil yang sama.
2. Membuat semua operasi kunci privat tidak bergantung pada input. Caranya misalnya dengan selalu melakukan tambahan pengurangan pada algoritma Montgomery walaupun hasilnya tidak kita pakai
3. Cara lain adalah memberikan sebuah delay sehingga setiap operasi yang sama memberikan waktu yang sama.

## IV. KESIMPULAN

*Timing attack* adalah sebuah Side channel attack yang menggunakan waktu untuk memecahkan kunci.

*Timing attack* mengeksploitasi kekuatan implementasi dari sebuah algoritma dari pada algoritmanya itu sendiri.

Semua algoritma kriptografi yang melibatkan operasi yang bisa diobservasi waktunya punya kemungkinan untuk diserang *timing attack*.

*Timing attack* sekarang telah berkembang dengan pesat, dan telah digunakan untuk memecah berbagai algoritma yang sulit.


## REFERENCES

- [1] Bar-El. Hagai, *Introduction to Side Channel Attacks*,. Netanya: Discretix, , pp. 2–4.
- [2] Paul C. Kocher: *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*. CRYPTO 1996, pp.104–113
- [3] H. Wong, Wing: *Timing Attacks on RSA: Revealing Your Secrets through the Fourth Dimension*

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Maret 2010



Sandy Socrates  
13508044